Address" property of the requester **108**. If they match, then the requester **108** is validated in step **145**. If the addresses do not match, an error **144** is issued. Step **146** involves verifying that the user still has rights to the form they are requesting. This is accomplished by issuing a NWDSGetEffectiveRights( ) of the use object **102** for the Forms Object **104** they have requested the NWDSGetEffectiveRights operation determines the access privileges of the user object **102** to the Forms object **104**, taking into account the user object's **102** security equivalences, inherited privileges, and inheritance masks. If step **147** determines that such rights exist, the method proceeds to step **149** where the rights are validated. If insufficient rights exist, an error **148** is issued.

In step **150**, the request is acted upon. A reference identifier to the Data Store **109**, in this case the employee ID attribute of the User object **102**, is read to reference appropriate information from the Data Store **109**. This reference identifier acts as a link between the identities in the Distributed Directory **101** and the Data Store **109**. In the case of a "Form Request", the form type is determined. If the form is a populated form, the Forms Processor Server **112** interfaces with the database **109** to read the appropriate data, populate the form, and send a reply e-mail for delivery to the user **108**. The completed form is stored in the GW.NPS.HTMLFORM field of the custom message. In the case of a "Form Submit", the Forms Processor Server **112** writes the submitted form data to the database **109** and sends back a response to the requester **108** indicating success or failure.

The Forms Processor Client **113** is implemented as a GROUPWISE Custom 3rd Party Object (C3PO), which will usually run on a client machine. This is an Ole Automation Server that is loaded by the GROUPWISE Client. When the GROUPWISE Client loads, it loads the Forms Processor Client **113**. The Forms Processor Client **113** is alerted that it has just been loaded and initiates it's own load process. FIG. **9** illustrates a flowchart **160** of steps performed by the Forms Processor Client **113** during the load process. In the step **161**, an entry is added to the "File" menu and a button is added to the tool bar in the GROUPWISE **111** user interface which allows the user **108** to invoke a "Form Request" dialog. At step **162**, the custom message type for the Forms Processor **112** is registered. Then a unique icon is associated with the new custom message type in step **163** to distinguish it from other message types in the InBox. At step **164**, the GROUPWISE data store (database schema) is extended to accommodate the following custom field definitions:

a. GW.NPS.ACT—alternate form identifier field.

b. GW.NPS.MSGDEF—field to hold the form name.

c. GW.NPS.SRCHKEY—field to hold the search key (employee id).

d. GW.NPS.FORMDATA—field to hold submitted form data.

e. GW.NPS.HTMLFORM—field to hold requested form data.

f. GW.NPS.REQUEST—field to hold the request type.

g. GW.NPS.REQUESTER—field to identify the requester/submitter.

h. GW.NPS.REQTARGET—field to identify request type (other/self).

i. GW.NPS.OTHERTARGET—field to identify request for other.

In step **165**, the Forms Processor Client **113** is in "Service" mode and waits for an appropriate command. When the "Form Request" dialog is invoked in step **166**, the Forms Processor Client **113** presents the user **108** with a list of all available forms on the "Form Request" dialog. This involves reading the current authenticated NDS object **102** "Forms" attribute, reading the "Forms" attribute of all the NDS groups the current user is a member, reading each container above the current authenticated object up to the [Root], and place all values in an internal object list. At step **168**, the user **108** simply double-clicks the form of interest on the list and the request is submitted in step **169** to the Forms Processor Server **112**. The Forms Processor Server **112** processes the request and responds with a custom message type, as discussed above.

The custom message is received in the GROUPWISE InBox as a unique message type represented by it's own icon. The form data is viewed by double-clicking the custom message in the InBox in step **170**. The Forms Processor Client **113** is alerted that a custom message it knows how to handle has been selected. At step **171**, the Forms Processor Client **113** reads all of the custom field values from the selected message. The GW.NPS.HTMLFORM data is saved to a local file in step **172**. Then, a custom viewer is launched in step **173**, which consists of an HTML browser for displaying the contents of the form. Once the form is loaded in the viewer, the form file from disk is deleted in step **174**. At step **175**, if the form contains a "Submit" button and if the button is clicked, the method will proceed to step **176** where the Forms Processor Client **113** submits the form to the Forms Processor Server **112** for processing. Step **176** involves retrieving the field names and field values from the form and placing them in the GW.NPS.FORMDATA custom field of the new submit message. Then, the values of the custom fields in the received message are placed in the custom fields of the new submit message. The new submit message is then sent to the Forms Processor Server **112**.

The computer system **100** provides the capability for NDS/GROUPWISE clients to request and submit data from a database **109** while leveraging NDS authentication and using the secure transport of GROUPWISE. A GROUPWISE client with the Forms Processor Client **113** loaded can request to request a form by clicking on the Request Form tool bar button. This launches the Request Form dialog box. Enumerated on this dialog box are the Form objects **104** the user **108** has been granted rights to request. This is accomplished by reading the Form associations from the current authenticated user object **102**, the groups this user is member of, and each of the containers above this user object up to [Root]. The user **108** can then request one of the enumerated forms from that dialog.

The NDS Form object **104** contains, as one of it's attributes, the address of the Forms Processor Server **112**. The GROUPWISE client uses this address to send the form to the Forms Processor Server **112**. The GROUPWISE client with the Forms Processor Server **112** is prepared to process any requests sent to it by requesting clients. Once a request is received, this puts the Forms Processor Server **112** into action. The received message custom fields are read, the request type is determined and the proper action to taken. If the Form type is "Populated Form", the Forms Processor Server interfaces with the database **109** to retrieve the appropriate data for the requesting user. A Form Type of "View" causes the Forms Processor Server **112** to interface with the database **109** and return a form with Read-only fields. An "Empty Form" type is returned immediately with no interface to the database **109** required.

The Forms Processor Server **112** sends the appropriate response to the Forms Processor Client **113**. The response message appears as a custom message in the GROUPWISE InBox. The user **108** can then double-click the message to